Consider the above code. We would like you to try to answer the questions without typing the code into a computer, but it might be a good idea to use a computer to double-check your answer.

1. What are the values of i, j, x and y at the point labeled “ONE”?  
   \( i=9, \ j=4, \ x=36.0, \ y=2.0 \)  (The .0s are not needed, but should be there)

2. What are the values of i, j, x and y at the point labeled “TWO”?  
   \( i=9, \ j=1, \ x=16.6, \ y=11.2 \)

3. What are the values of i, j, x and y at the point labeled “THREE”?  
   \( i=3, \ j=2, \ x=17.0, \ y=5.0 \)

4. Which, if any, of your answers would be different if i and j were declared as doubles?  
   In #1 y would be 2.25  
   In #2 j would be 2.5454….  
   In #3 y would be 4.3333333  
   (The variables i and j might also be written with a .0 following them.)  
   Finally the question only asked which would change, so a list without values would be fine.
Part II (20 points)

```cpp
#include<iostream>
using namespace std;

int slow_m(int can, int plier)
{
    int i=0;
    int sum=0;
    while(i<can)
    {
        sum=sum+plier;
        i=i+1;
    }
    return(sum);
}

int slow_d(int d1, int d2)
{
    int count=0;
    int total=d2;
    while(total>d1)
    {
        total=total-d1;
        count=count+1;
    }
    return(count);
}

int slow_mod(int base, int div)
{
    int count=0;
    int total=base;
    while(total>div)
    {
        total=total-div;
        count=count+1;
    }
    return(total);
}

main()
{
    int a;
    a= slow_m(4,8);
    a= slow_m(-3,4);
    a= slow_m(4,-3);
    a= slow_d(12,3);
    a= slow_d(3,12);
    a= slow_d(-5,30);
    a= slow_mod(12,3);
    a= slow_mod(3,12);
    a= slow_mod(-5,12);
}
```

Answer the following questions about the code:

1. What value is returned by the calls to slow_m()?
   a. slow_m(4,8)= __32___
   b. slow_m(-3,4)= __0___
   c. slow_m(4,-3)= __-12___

2. When slow_d(12,3) is called, what is the value of total when the function exits?
   **Total = 3**

3. One of the slow_d() invocations will “run forever” (not really, but from what we know it will). Which one and why?
   **slow_d(-5,30). Total will never be greater than d1.**

4. When slow_d(3,12) is called, what is the value of total at the instant count becomes 2?
   **Total = 6**

5. What value is returned by the calls to slow_mod()?
   a. slow_mod(12,3) = __3___
   b. slow_mod(3,12) = __3___
   c. slow_mod(-5,12) = __-5___
1. What will be the value of the variable “stepsize”? Why do you think it has been declared as “const”? 
   It is 1.0. It is declared const because it is a constant and won’t change. (Also, one might notice it is made up only of constant values.)

2. What will be the initial value of the variable “step”? ____0.5____

3. What will be the value of “step” at the instant the variable “i” is assigned the value 2? ____1.5____

4. If COUNT were instead set to be 2, what would be the value printed by this program? ___312.5___
   (This one won’t be counted, but it is useful bit of practice)

5. How would you change this program if you wanted a reasonably accurate value for \( \int_{3}^{12} \left(\frac{x}{x + 1}\right) \)? Be specific about exactly how you would change specific lines of code.
   Change the COUNT to be a larger number. Say 10000. Next bottom=3.0 and top=12.0. Finally change the function to return(x/(x+1)). Not bad for a short little program.